

Fig. 1

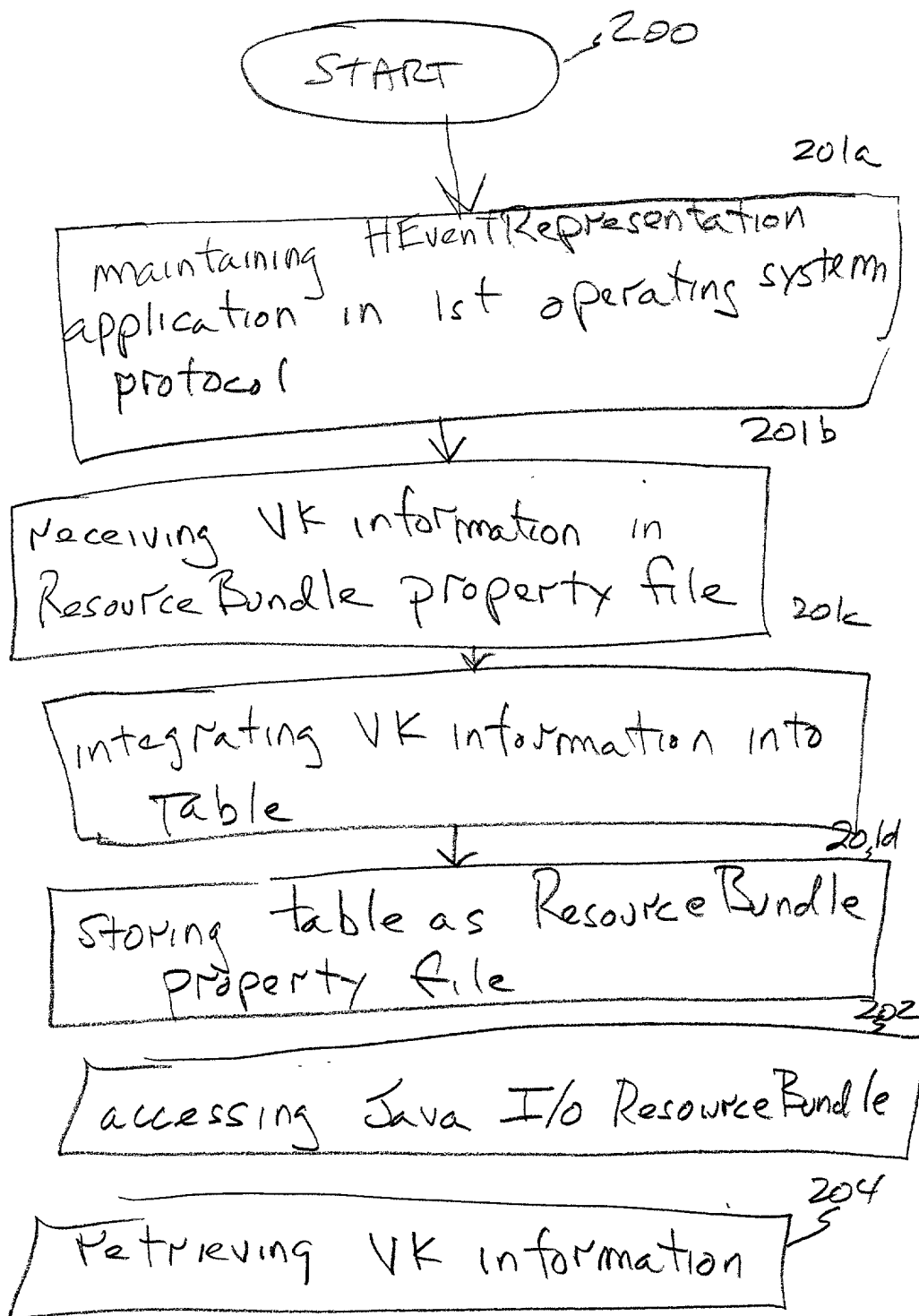


Fig. 2

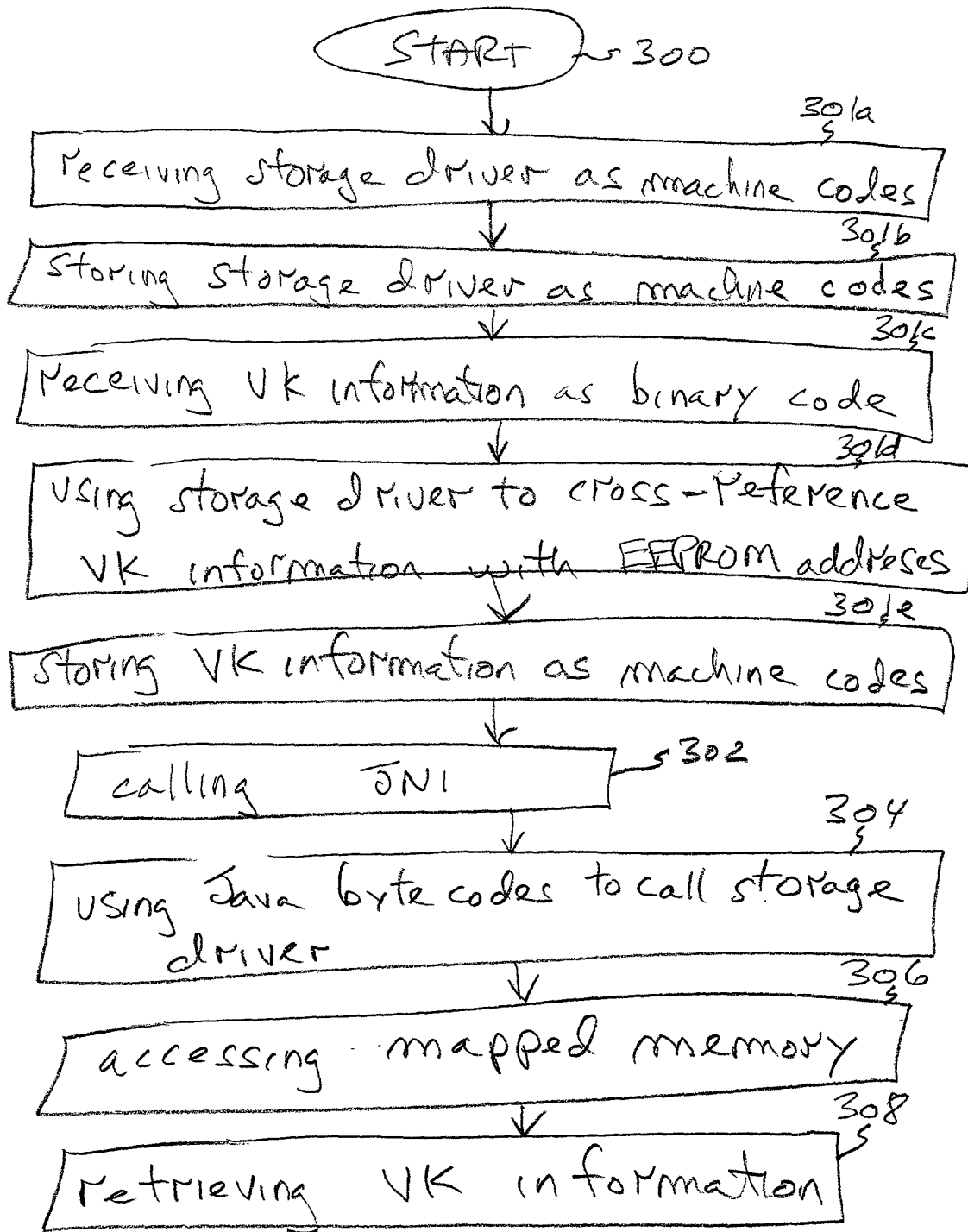


Fig. 3

```

**platform event representation - color]
VK_GO_TO_START_Color = 21, 21, 21
VK_REWIND_Color = 0, 0, 0
VK_STOP_Color = 0, 0, 0
VK_PAUSE_Color = 0, 0, 0
VK_PLAY_Color = 0, 0, 0
VK_FAST_FWD_Color = 0, 0, 0
VK_GO_TO_END_Color = 0, 0, 0
VK_TRACK_PREV_Color = 0, 0, 0
VK_TRACK_NEXT_Color = 0, 0, 0
VK_RECORD_Color = 245, 0, 0
VK_EJECT_TOGGLE_Color = 0, 0, 0
VK_VOLUME_UP_Color = 0, 0, 0
VK_VOLUME_DOWN_Color = 0, 0, 0
VK_UP_Color = 0, 0, 0
VK_DOWN_Color = 0, 0, 0
VK_LEFT_Color = 0, 0, 0
VK_RIGHT_Color = 0, 0, 0
VK_POWER_Color = 0, 0, 0

**platform event representation - String]
VK_GO_TO_START_String = Two equilateral triangles, pointing at a line to the left
VK_REWIND_String = Two equilateral triangles, pointing to the left
VK_STOP_String = A square
VK_PAUSE_String = Two vertical lines, side by side
VK_PLAY_String = One equilateral triangle, pointing to the right
VK_FAST_FWD_String = Two equilateral triangles, pointing to the right
VK_GO_TO_END_String = Two equilateral triangles, pointing to a line at the right
VK_TRACK_PREV_String = One equilateral triangle, pointing to a line at the left
VK_TRACK_NEXT_String = One equilateral triangle, pointing to a line at the right
VK_RECORD_String = A circle, normally red
VK_EJECT_TOGGLE_String = A line under a wide triangle which points up
VK_VOLUME_UP_String = A ramp, increasing to the right, near a plus sign
VK_VOLUME_DOWN_String = A ramp, increasing to the right, near a minus sign
VK_UP_String = An arrow pointing up
VK_DOWN_String = An arrow pointing down
VK_LEFT_String = An arrow pointing to the left
VK_RIGHT_String = An arrow pointing to the right
VK_POWER_String = A circle, broken at the top, with a vertical line in the break

**platform event representation - image]
VK_GO_TO_START_Image = start.png
VK_REWIND_Image = rewind.png
VK_STOP_Image = stop.png
VK_PAUSE_Image = pause.png
VK_PLAY_Image = play.png
VK_FAST_FWD_Image = fastfwd.png
VK_GO_TO_END_Image = end.png
VK_TRACK_PREV_Image = prevtrack.png
VK_TRACK_NEXT_Image = nexttrack.png
VK_RECORD_Image = record.png
VK_EJECT_TOGGLE_Image = eject.png
VK_VOLUME_UP_Image = volup.png
VK_VOLUME_DOWN_Image = voldown.png
VK_UP_Image = up.png
VK_DOWN_Image = down.png
VK_LEFT_Image = left.png
VK_RIGHT_Image = right.png
VK_POWER_Image = power.png

**platform event representation - type]
VK_GO_TO_START_Type = ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_REWIND_Type = ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_STOP_Type = ER_TYPE_SYMBOL
VK_PAUSE_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_PLAY_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_FAST_FWD_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_GO_TO_END_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_TRACK_PREV_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_TRACK_NEXT_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_RECORD_Type = ER_TYPE_SYMBOL
VK_EJECT_TOGGLE_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_VOLUME_UP_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_VOLUME_DOWN_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_UP_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_DOWN_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_LEFT_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_RIGHT_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL
VK_POWER_Type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL

```

Fig. 4a Property text file to store HEventRepresentation table.

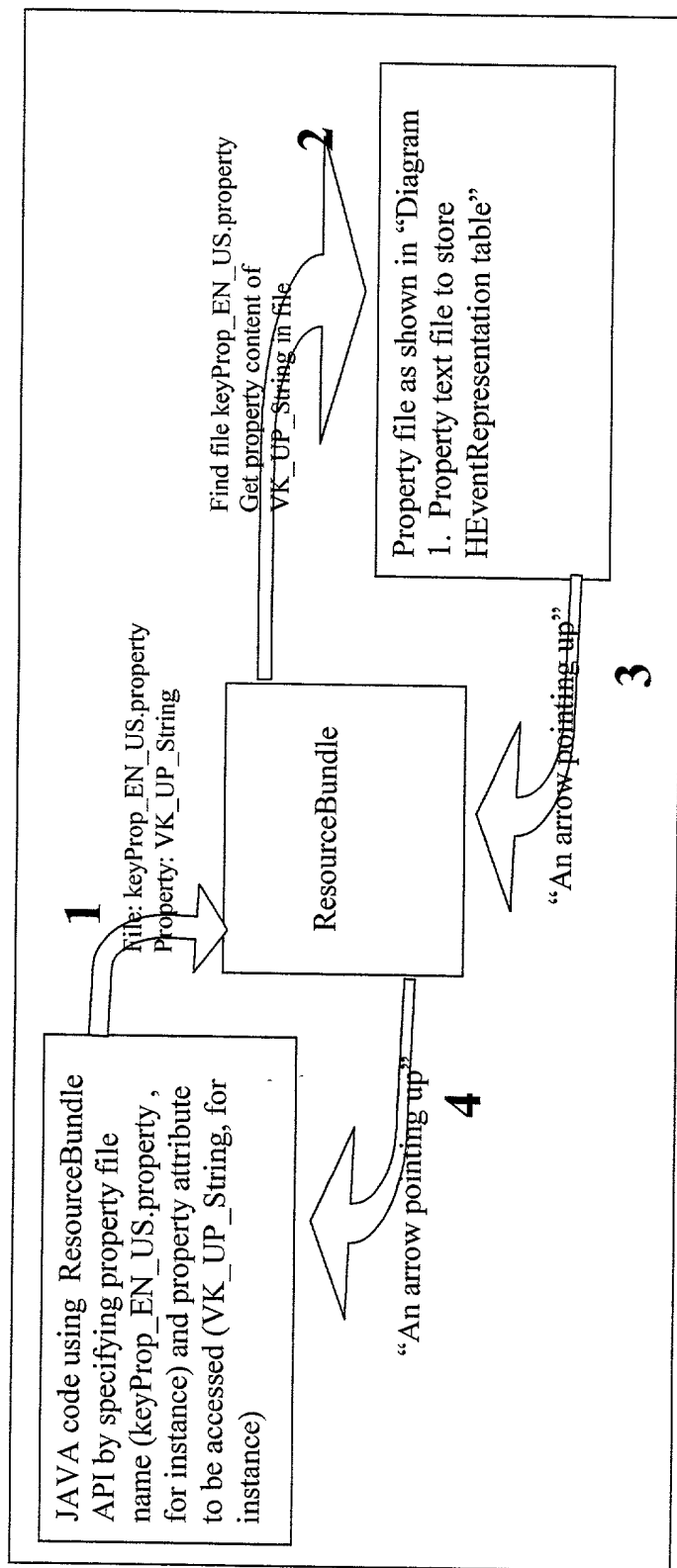


Fig. 4b

```
String[] eventRepresentationData = {
    VK_GO_TO_START, new Color(r, g, b), "⏮", "start.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_REWIND, new Color(r, g, b), "⏮", "rewind.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_STOP, new Color(r, g, b), "STOP", "stop.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_PAUSE, new Color(r, g, b), "⏸", "pause.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_PLAY, new Color(r, g, b), "▶", "play.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_FAST_FWD, new Color(r, g, b), "⏭", "fastfwd.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_GO_TO_END, new Color(r, g, b), "⏭", "end.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_TRACK_PREV, new Color(r, g, b), "⏮", "prevtrack.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_TRACK_NEXT, new Color(r, g, b), "⏭", "nexttrack1.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_RECORD, new Color(r, g, b), "O", "record.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_EJECT_TOGGLE, new Color(r, g, b), "EJECT", "eject.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_VOLUME_UP, new Color(r, g, b), "VOL+", "volup.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_VOLUME_DOWN, new Color(r, g, b), "VOL-", "voldown.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_UP, new Color(r, g, b), "▲", "up.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_DOWN, new Color(r, g, b), "▼", "down.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_LEFT, new Color(r, g, b), "◀", "left.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_RIGHT, new Color(r, g, b), "▶", "right.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
    VK_GO_TO_POWER, new Color(r, g, b), "0/1", "nght.png", ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL,
};
```

Fig. 5 Text array of key event representation

```

package com.sharplabs.hav;
import java.awt.Color;

public static class EventRepresentationData {
    static class VK_GO_TO_START {
        static final int code = VK_GO_TO_START;
        static final Color c = new Color(0, 0, 0);
        static final String s = "<<";
        static final String imgFile = "start.png";
        static final int type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL;
    }

    static class VK_REWIND {
        static final int code = VK_REWIND;
        static final Color c = new Color(0, 0, 0);
        static final String s = "<<";
        static final String imgFile = "rewind.png";
        static final int type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL;
    }

    static class VK_STOP {
        static final int code = VK_STOP;
        static final Color c = new Color(0, 0, 0);
        static final String s = "stop";
        static final String imgFile = "stop.png";
        static final int type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL;
    }

    static class VK_PAUSE {
        static final int code = VK_PAUSE;
        static final Color c = new Color(0, 0, 0);
        static final String s = "|";
        static final String imgFile = "pause.png";
        static final int type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL;
    }

    static class VK_PLAY {
        static final int code = VK_PLAY;
        static final Color c = new Color(0, 0, 0);
        static final String s = ">>";
        static final String imgFile = "play.png";
        static final int type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL;
    }

    static class VK_FAST_FWD {
        static final int code = VK_FAST_FWD;
        static final Color c = new Color(0, 0, 0);
        static final String s = ">>";
        static final String imgFile = "fast_fwd.png";
        static final int type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL;
    }

    static class VK_GO_TO_END {
        static final int code = VK_GO_TO_END;
        static final Color c = new Color(0, 0, 0);
        static final String s = ">>";
        static final String imgFile = "end.png";
        static final int type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL;
    }

    static class VK_TRACK_PREV {
        static final int code = VK_TRACK_PREV;
        static final Color c = new Color(0, 0, 0);
        static final String s = "<";
        static final String imgFile = "prevtrack.png";
        static final int type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL;
    }

    static class VK_TRACK_NEXT {
        static final int code = VK_TRACK_NEXT;
        static final Color c = new Color(0, 0, 0);
        static final String s = ">";
        static final String imgFile = "nexttrack.png";
        static final int type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL;
    }

    static class VK_RECORD {
        static final int code = VK_RECORD;
        static final Color c = new Color(0, 0, 0);
        static final String s = "REC";
        static final String imgFile = "record.png";
        static final int type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL;
    }

    static class VK_EJECT_TOGGLE {
        static final int code = VK_EJECT_TOGGLE;
        static final Color c = new Color(0, 0, 0);
        static final String s = "EJECT";
        static final String imgFile = "eject.png";
        static final int type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL;
    }

    static class VK_VOLUME_UP {
        static final int code = VK_VOLUME_UP;
        static final Color c = new Color(0, 0, 0);
        static final String s = "VOL+";
        static final String imgFile = "volumeup.png";
        static final int type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL;
    }

    static class VK_VOLUME_DOWN {
        static final int code = VK_VOLUME_DOWN;
        static final Color c = new Color(0, 0, 0);
        static final String s = "VOL-";
        static final String imgFile = "volumedown.png";
        static final int type = ER_TYPE_NOT_SUPPORTED | ER_TYPE_STRING | ER_TYPE_COLOR | ER_TYPE_SYMBOL;
    }

    static class VK_UP {

```

Fig. 6 Static class of event representation

Applications

HAVIL2 GUI API

Table storage in JAR file in
the model of either
- static class
- data array
(Storage media, e.g., ROM)

Via java virtual machine operation

Fig. 7 JVM Access Model

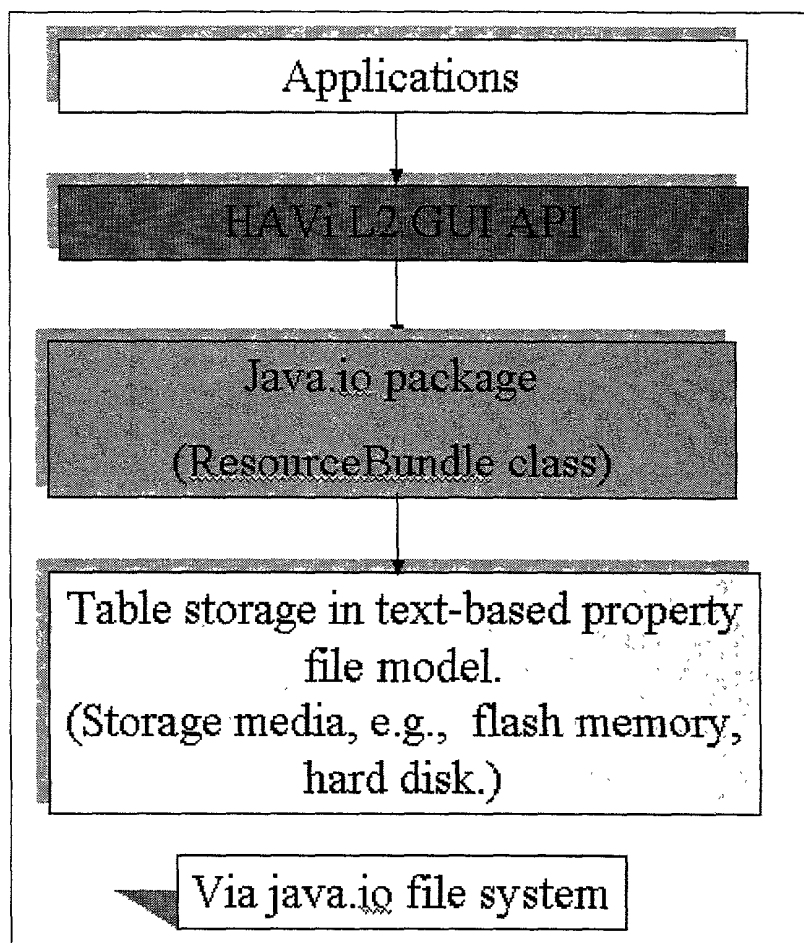


Fig. 8 Java I/O Accessing model

Applications

HAVIL2 GUI API

JNI calls to storage device driver
(each event code, color, string,
symbol representation maybe in
binary format)

Table storage in mapped memory
(Storage media, e.g., EEPROM)

Via JNI / device drivers

Fig. 9 JNI/Storage driver Accessing Model

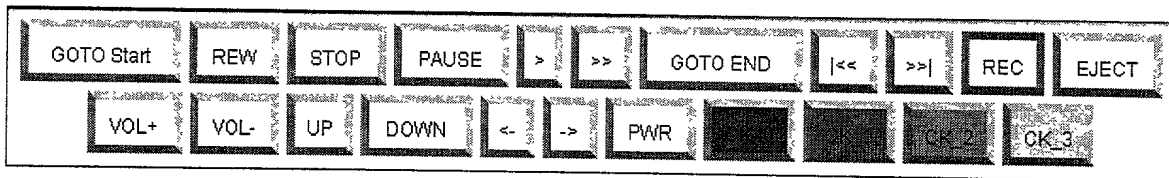


Fig. 10 HEventRepresentation using String, Color attribute data

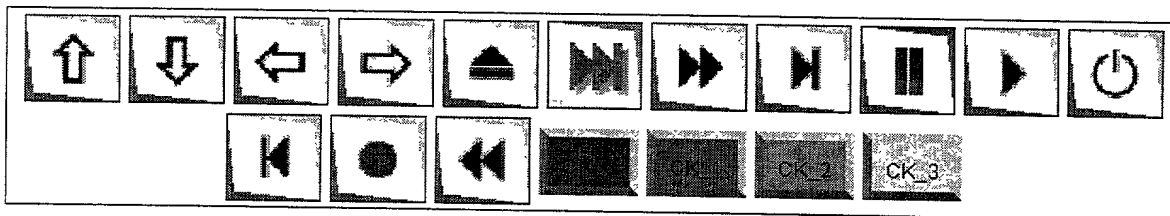


Fig. 11 HEventRepresentation using Symbol, String, and Color attribute data